Aufgabe: Eine LED soll leuchten, wenn es dunkel wird bzw. wenn ein Fotowiderstand (Sensor für Helligkeit) abgedeckt wird.

Material:

- Arduino-Board
- Eine blaue LED
- Ein Widerstand mit 100 Ohm
- Ein Widerstand mit 10K Ohm (10000 Ohm)
- Breadboard
- Steckkabel
- Ein Fotowiderstand (Helligkeitssensor)

In dieser Anleitung lernst du:

- Analoge Sensorwerte auslesen (Spannung am Analogpin auslesen)
- Analoge Sensorwerte verwenden
- Variablen für Sensorwerte nutzen
- Fotowiderstand anschließend
- Fotowiederstand situationsgerecht einsetzen
- Serial Monitor zur Anzeige von Messwerten nutzen

Spannungen auslesen:

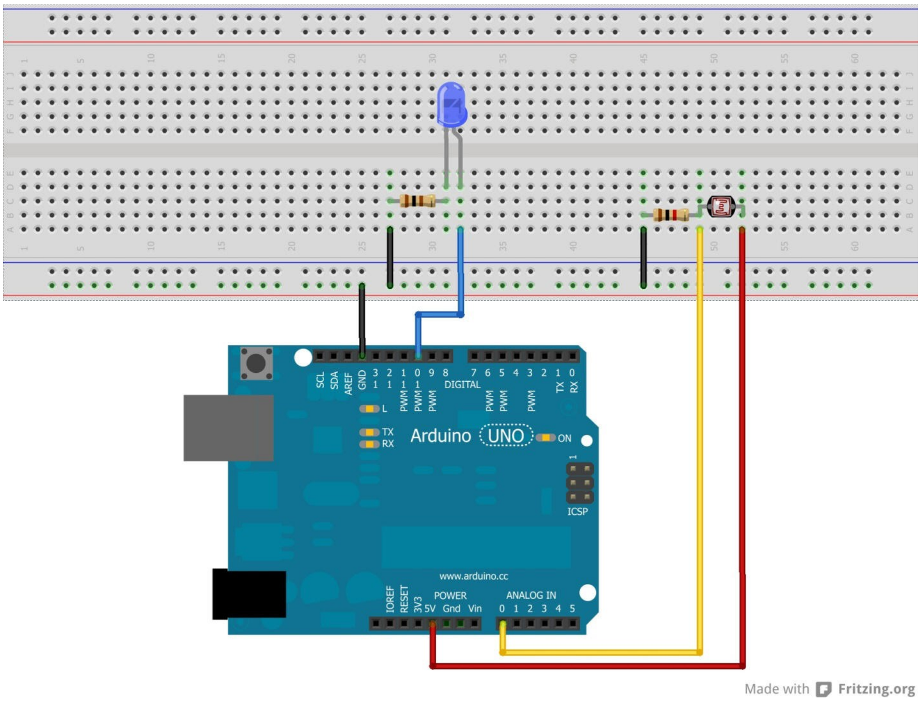
Der Mikrokontroller soll über einen Fotowiderstand auslesen, wie hell es ist. Dazu nutzt man ein einfaches physikalisches Prinzip. Wenn in einem Stromkreis zwei Verbraucher hintereinander angeschlossen sind (Reihenschaltung), dann „teilt" sie sich auch die gemeinsam anliegende Spannung. Ein Beispiel: Zwei gleiche Lampen sind in Reihe geschaltet und es wird eine Spannung von 6 Volt angelegt. Dann kann man mit einem Spannungsmessgerät feststellen, dass an den Lampen jeweils nur 3 Volt anliegen. Wenn zwei ungleiche Lampen angeschlossen werden (Eine hat einen geringeren Widerstand), dann kann man zwei unterschiedliche Spannungen an den beiden Lampen messen, bspw. 1,5 Volt und 4,5 Volt. Ein Fotowiderstand ändert seinen Widerstand in Abhängigkeit der Lichtstärke. Diesen Effekt nutzt man, um anhand der an ihr anliegenden Spannung einen Wert für Helligkeit bzw. Dunkelheit in Form von verschiedenen Spannungen abzulesen. Damit man hier überhaupt eine Spannungsteilung erzeugen kann, schließt man den Fotowiderstand und einen Widerstand (1 – 10 K Ohm, je nach verwendetem Fotowiderstand. Der Widerstand sollte einen ähnlichen Widerstandswert wie der Fotowiderstand haben) in Reihe an und verbindet sie wie in der Skizze mit 5 Volt und der „Erdung" (Ground / GND). Das Mikrocontroller-Board ist in der Lage, analoge Signale (Spannung) zu messen und diese zu verarbeiten. Dies geschieht mit den analogen Eingängen auf dem Board. Dieser wandelt den gemessenen Spannungswert in eine Zahl um, die dann weiter verarbeitet werden kann. 0 Volt entspricht dabei dem Wert 0 und der höchste Messwert bei 5 Volt entspricht dem Wert 1023 (0 bis 1023 entspricht 1024 Zahlen = 10 Bit). Beispiel: Es wird eine Spannung von 2,5 Volt gemessen, dann liefert der Mikrokontroller den Wert 512 (1024 : 2).

Der „serial monitor":

Der „serial monitor" ist ein wichtiger Bestandteil der Arduino-Software. Mit diesem „serial monitor" kann man sich am PC Daten anzeigen lassen, die das Mikrocontroller-Board an den PC sendet (Zahlen oder Texte).

Das ist sehr sinnvoll, da man nicht immer ein LCD Display am Mikrocontroller angeschlossen hat, auf dem man bestimmte Werte ablesen könnte. In diesem Sketch wird der „serial monitor“ verwendet, um die Werte anzeigen zu lassen, die das Board von dem Fotowiderstand einliest. Wozu ist das sinnvoll? Mal angenommen, die LED soll erst bei beginnender Dunkelheit anfangen zu leuchten. Dann muss es im Sketch einen Bereich geben, der die Funktion hat: „Wenn der Wert des Fotowiderstandes den Wert x unterschreitet, dann soll die LED leuchten“. Dazu müsste man wissen wie groß der Wert x bei beginnender Dämmerung ist.
Lösung: Ich sende den ausgelesenen Wert „x“ der Spannung an dem Fotowiderstand bei entsprechender Helligkeit (bspw. Dämmerung) an den „serial monitor“ und lasse ihn mir dort anzeigen. Mit diesem Wissen kann ich später das Programm in der folgenden Form abändern. „Wenn der Spannungsausgabewert des Fotowiderstandes einen Wert von „x“ unterschreitet, dann schalte die LED an.“

Aufbau:



Sketch:

```
int eingang=A0;                  //Das Wort „eingang“ steht jetzt für den Wert „A0“
                                 (Bezeichnung vom Analogport 0)
int LED=10;                      //Das Wort „LED“ steht jetzt für den Wert 10
int sensorwert=0;                //Variable für den Sensorwert mit 0 als Startwert

void setup()                     //Hier beginnt das Setup.
{
Serial.begin(9600);              //Die Kommunikation mit dem seriellen Port wird
                                 gestartet, um den Wert des Sensors im serial
                                 monitor darzustellen.
pinMode (LED,OUTPUT);            //Der Pin mit der LED (Pin 10) ist jetzt ein
                                 Ausgang
}

void loop()
{                                //Mit dieser Klammer wird der Loop-Teil geöffnet.
sensorwert=analogRead(eingang); //Die Spannung an dem Fotowiderstand auslesen
                                  und unter der Variable „sensorWert“ abspeichern.
Serial.print("Sensorwert=");     //Ausgabe am Serial-Monitor: Das Wort
                                 „Sensorwert:”
Serial.print(sensorwert);        // Ausgabe am Serial-Monitor. Mit dem Befehl
                                 Serial.print wird der Sensorwert des
                                 Fotowiderstandes in Form einer Zahl  zwischen 0 und
                                 1023 an den serial monitor gesendet.
if(sensorwert > 512)             //Wenn der Sensorwert über 512 beträgt….
{
digitalWrite(LED,HIGH);          //…soll die LED leuchten…
}
else
{
digitalWrite(LED,LOW);           //…andernfalls soll sie nicht leuchten
}
delay(50);                       //Eine kurze Pause, in der die LED an oder aus ist
}                                //Mit dieser letzten Klammer wird der Loop-Teil
                                 geschlossen.

//Wenn nun der Sensorwert bei normaler Helligkeit bspw. nur den Wert 100 hat
(Der Wert ist abhängig von den verwendeten Widerständen, von der Helligkeit und
von der Stromrichtung), dann nimmt man anstelle des Wertes 512 einen wesentlich
kleineren Wert, bei dem die LED zu leuchten beginnen soll. Bspw. nimmt man dann
den Wert 90. Den aktuellen Sensorwert kann man sich nun mit Hilfe des „Serial
monitor“ anzeigen lassen. Dazu klickt man oben auf „Werkzeuge“ bzw. „Tools“ und
anschließend auf „serial monitor“.
```